ACE USER

NEWSLETTER OF THE JUPITER ACE USERS CLUB

no.2 Spring 1983

DEAR ACE USER,

The good news is that the national Jupiter Ace Users Club is growing - and growing fast. Now over a hundred members as I write, and probably nearer two hundred by the time you read this, the Club looks set for a bright future.

We hope for a high standard with our newsletter, and are reasonably satisfied with no.2. We hope you are too! Contributions are very welcome - we can pay £10 for articles similar to those by Doug Bollen in this issue.

The software catalogue has built up quickly with some excellent tapes, and more to follow. We are particularly interested in programs which strengthen this range, eg. mathematical, educational, and graphics. Now the 16K and 32K add-on memories are available, programmers should make use of this extra memory wherever possible. Royalties of 20% of sale price are paid (and paid promptly every quarter).

We presume the lack of promised add-ons from Jupiter Cantab indicates a cash-flow problem. However add-on manufacturers are beginning to fill the gap. Stonechip have provided 16K and 32K memory (available through the Club), and have indicated that an Ace version of their Echo unit for the Spectrum (amplified sound and improved tape handling) could be produced if there is sufficient demand (let us know!). Innovonics have plans in this field, and as we go to press an I/O port has been advertised by Haven Hardware.

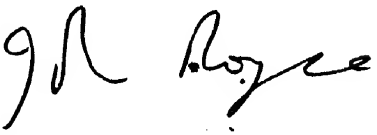
We ourselves have produced the adaptor kit for using ZX81 add-ons (memory, and sound units, etc). Our future plans include an expansion board to take ROM chips, a printer interface board with I/O port (still no sign of Jupiter Cantab's promised serial/parallel printer board) and possibly a music synthesiser board.

The ZX printer looks like it requires a fiendish piece of machine code to be of use, and is probably not worth using (anyone care to prove us wrong?).

Several Ace users have enquired about linking up a proper keyboard. If anyone has done this, an article for the next issue would be appreciated.

The smaller of the two outputs at the back of the Ace is apparently for a planned colour board - but again no sign of one from Jupiter Cantab. Has anyone experimented with this output?

Letters and queries are always welcome (SAE for immediate answer, otherwise reply in next newsletter)



JOHN NOYCE, NEWSLETTER EDITOR

ACE EXPANSION

DOUG BOLLEN

One of the attractive features of the ACE is its ability to use inexpensive ZX81 peripherals. Also, the quick LOAD/SAVE makes large Ram expansion a worthwhile proposition. What a pity that JUPITER CANTAB did not copy the ZX81 edge connector layout!

RAMPACKS

Tests and reports indicate that 16K, 32K and 48K ZX Rampacks will run on the ACE, via a suitable adaptor, without modification (yielding Ramtops of 32768, 49152, and 65536 respectively).

If you include the 3K of Ram below 16384 claimed by JUPITER CANTAB, you then have user memories of 19K, 35K, and 51K. The bigger ZX81 Rampacks containing 56K (sometimes designated as 64K) use special addressing techniques to reclaim the unused 8192-16384 area above the ZX81 Rom, as do Eprom/Cmos packs, and these cannot be expected to work on the ACE without modification. Some 56/64K Rampacks have switches to cut out the 8K to 16K area, so these should work on the ACE.

Non-volatile Ram, extra Rom, and Eprom are probably best situated above Ramtop on the Ace, within the 49152-65536 addressed area. You could employ a 32K Rampack plus a card containing up to 16K of Eprom/Rom/Cmos.

CRASHING

What about Ram wobble and white-outs? The ZX81 format of edge connector is far from ideal, and this is - unfortunately - perpetuated on the ACE. There is a lack of gold plating on the male(circuit board) connector, which leads to oxidisation and high contact resistance, and locking lugs are omitted.

Make sure your connectors are tight, bright and clean, Two methods of tightening the fit are building up the circuit board contacts with solder (solder is a rotten contact material though) or very carefully prising the gold-plated spring contacts on the female connector with something like a thick sewing needle.

The next step in the reliability stakes is a major choice between two options. Do you want your add-ons strapped to the computer case with gaffer tape, sticky pads, or similar adhesive methods, so that the vibration of rubber keyboard pounding is not transmitted to the edge connector, or do you prefer a flexible connection to a bevy of spaghetti linked circuit boards? Some people have an aversion to wires and will choose to fashion their peripherals into one solid block with the computer, while others prefer the flexibility of easily altered wire systems.

The ZX81 uses resistors in series with its data bus, which severely limits the linking of peripherals by lengthy ribbon cable unless a buffering circuit is used, but the ACE seems to be better in this respect. One 16K Rampack we tried refused to run for more than a few minutes when plugged straight into a ZX81, but worked perfectly on the ACE at the end of 15cm of flexible cable!

With a flexible connector, it should be possible to move the Rampack several cms without crashing.

Here is a good Rampack reliability test,

```
16384 allot
: test
  begin
  15384 @ u. 28 spaces
  0
  until
;
```

This shifts the test routine 16K up in memory and then scrolls the value of Ramtop continuously. Leave the ACE running for several days and nights, and try tapping the Rampack occasionally. If it has not crashed at the end of the period your edge connectors are good and there are no frequent mains glitches in your vicinity. Watch out for fridges and freezers, with their nasty switch-on surges. CB radio is also reputed to cause problems.

POWER SUPPLIES

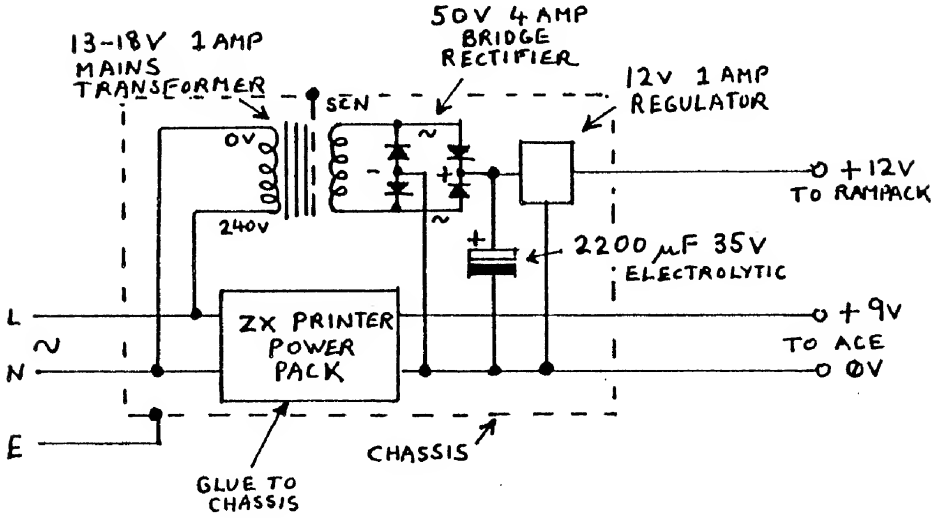
If your Rampack will not run reliably with good edge connectors and no mains glitches, the trouble could lie with your power supply. Many a crash falsely attributed to edge connectors is caused by power supply deficiencies.

Rampacks in the bottom price bracked often undervoltage their 4116 ICs to cut corners, and although they may return the correct value for Ramtop, operation within the expanded area of Ram can be unpredictable, particularly when stacking add-ons.

The unexpanded ACE consumes around 700 mA at 9.5V, near the limit of its power supply. If you take, say, 200 - 300 mA more, the power pack will get hotter and the voltage will drop below 9V. 4116 chips prefer 12V. . JUPITER CANTAB are now issuing a bulkier power pack which is not integral with the mains plug. Although rated the same as model 1, it may have more current to spare. Using the ZX printer power pack will improve matters, but the Ram will still be undervoltaged. The only cure is to cut the track between the Rampack and the 9V rail on its edge connector, and take +12V via a flying lead to the 4116s. The circuit we used successfully is shown in Fig 1.

To check that your 4116s are, in fact, undervoltaged, take the Ram circuit board out of its case and power up. Very carefully - to avoid devastating short-circuits - apply the prods of a voltmeter to both leads of one of the disc ceramic capacitors mounted near the end of each 4116. Reverse the prods if the polarity is wrong. If your reading is within 11 - 13V, with and without extra add-ons, forget about a special 12V power supply.

Fig 1



UPGRADED ACE POWER SUPPLY

Note the absence of an earth on the 0V rail (normal with ZX81 and ACE power packs), this helps to combat mains glitches and hum loops. Don't be worried about safety. With modern insulation techniques, earthing of plastic cased equipment is not mandatory, but make sure that the mains transformer metalwork and power supply chassis is well earthed.

PORTS

There have been no serious hardware problems encountered thus far with other ZX81 add-ons interfaced to

the ACE. The only proviso is that addressing does not conflict with existing Ram. Memory mapped ports (usually located near 32768) can be accessed like Ram using the techniques given in the manufacturers' instructions, with the FORTH equivalents of PEEK and POKE (c@ and c!).

Z80 in-out configured ports are particularly suited to the ACE as they will - in almost all cases - respond to the FORTH IN and OUT words, without additional machine code, and without conflicting with Ram. The William Stuart CHATTERBOX speech synthesiser, for example, needs a machine code routine and a string array to work on the ZX81, but functions quite happily with FORTH words only on the ACE. If readers are interested in using CHATTERBOX with the ACE we can give software details in the next issue.

ACE IN OUT commands use single byte addressing, which limits port addresses to all odd numbers between 0 and 256. When IN and OUT take a two byte number off the FORTH stack, they discard the hi-byte. If you want to manipulate single bytes on the FORTH stack it is important to remember that the h-byte is on top. 15419 @ 1 - c@ . gives the hi-byte and 15419 @ 2 - c@ . gives the lo-byte. The manual (p143) misleads; 15419 @ does give the first address past the top of the stack.

If your Z80 in-out configured port mysteriously refuses to respond to FORTH IN or OUT words, as one we tried did, you can use a short machine code routine with out (port address), A and in A, (port address) instructions.

The fast ACE is particularly suited to driving music synthesiser ICs like the AY - 3 - 8910, because it can shape ADSR envelopes for short duration notes without machine code, which is not possible in ZX BASIC.

The ZX printer is merely a Z80 in-out configured port, but it does need precise software to run it. We have been looking at this and hope to produce an adaptable FORTH printer routine which can be incorporated in your program (when time permits!).

The ACE load/save port resides at address 254 (FE) and it is a simple matter to access it with IN and OUT. You can send signals to your tape recorder or hi-fi with

248 254 OUT

and Ø 254 OUT which switches the port on and off. Receive signals with

254 IN which leaves a number on the stack, and also responds to key presses.

Here is a simple routine for displaying a program on tape as a block of black and white spaces scrolling up the screen, with tape noise and clicks represented by sporadic inverse spaces:

```
: TEST
  BEGIN
  254 IN
  63 <
  IF
  ." □"
  ELSE
    SPACE
  THEN
  Ø
  UNTIL
;
```

try pressing the keys whilst TEST is running.

COMPUTER MODIFICATIONS

If you don't know how to extract the plastic rivets on your ACE yoghourt pot case (we hear a rumour that JUPITER CANTAB are going to use a stronger injection moulded case in the future), just push the centre of the rivet with a thin blunt spike. The plastic retaining pin will drop inside the case, allowing the rivet to be pulled out. The circuit board is held to the lower half of the case by three similar rivets. Of course, we do not suggest that you open up your ACE, if you are not an electronics buff and are not prepared to sacrifice your guarantee.

The ACE BEEP sound level is pathetic but JUPITER CANTAB have hinted that you can do a five minute rewire job on the computer circuit board to get loud sound from your TV. if you have done this modification please send details to ACE USER, before we all go out and buy deaf aids!

In the meantime, you can get a loud sound from the tape recorder with

```
: sound
  5ØØ Ø do
  248 254 out
  Ø 254 out
  loop
;
```

which cycles the SAVE (mic) port on and off for about 1 second. It is possible to use the LOAD and SAVE port as a serial interface, and the word sound could be developed to generate a range of sound effects, crude speech or even send data to another ACE or a SPECTRUM. In machine code, the instruction out (254), A where A contains any number between 248 and 255 will switch the LOAD port on, and contents of A between Ø - 247 will switch it off. Alternatively, you can toggle bit 3 with the other bits on.

If any ZX81 peripheral hardware manufacturer would like to lend us their product to test on the ACE we would be glad to publish details as part of our service to members,

but do include full operating instructions and check that the unit works on a ZX81 first. Similarly, if readers have been interfacing with their local power station, or anything else, we will be pleaseed to hear about it.

Finally, to end, let me leave you with this thought. Imagine an ACE with colour add-on and hi-resolution graphics board running a 51K program in FORTH! Good luck with your expansions!

THE OUTPUT PORT FOR THE ACE

The following contribution comes from an unnamed engineer with the Spanish agents for the ACE. Our thanks to Dante Smith for the translation.

In the circuit p152 of the ACE manual, we think there may be errors for the following reasons.

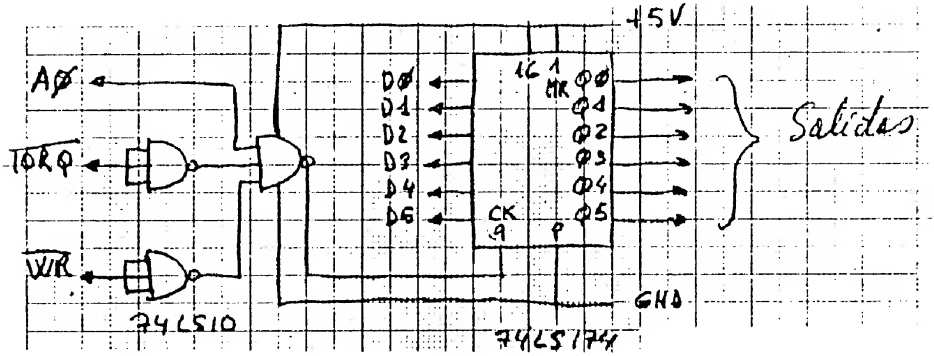
In the manual line A1 of the address bus is used. This makes it impossible to use address 1 to address the port, nor can it be an odd number because decimal 1 is equal to binary 1 and leaves logic 1 at line AØ.

On the other hand, the conditions necessary so that the port can accept valid data is

$\overline{IORQ} = Ø$, $\overline{WR} = Ø$ and AØ = 1

(see the time diagram of the Z80A CPU in Zilog's manual).

In these conditions a single gate NOR with 3 inputs cannot control them for the above reasons. We think the following diagram is what is needed:



The circuit suggested in the manual caused problems, and this new circuit works perfectly.

We havent tried this circuit, but we think the Senor is correct. The same probably applies to the circuit diagram on p154 of the manual as well. An OOOOPS STEVEN?

Apparently the Spanish agents for the ACE have gone to the trouble of translating the entire manual into Spanish, and putting in additional program examples ... so those of you disatisfied with the manual (and who understand Spanish of course) know where to look! Perhaps if the English ACE manual ever runs to a third edition we can look forward to the Spanish contributions appearing in English?
Meanwhile if our Spanish colleagues would like to contribute their additional material, with English translation, we'll be pleased to publish it in ACE USER. John

Why use machine code at all if you have fast FORTH? A good question. FORTH permits laborious-to-program machine code to be avoided in all but a few cases. Here is one.

Suppose you followed the suggestion in your manual and have defined CMOVE (see page 130):

```
: CMOVE   (address 1, address 2, bytes - )
  DUP IF ELSE EXIT THEN          (check for zero bytes)
  SWAP DUP ROT 1+ + SWAP         (set up DO LOOP)
  DO
  DUP c@ 1 C! 1+                 (transfer bytes)
  LOOP
;                      (memory cost 64 bytes)
```

You shift 100 bytes, fine! Now you stick on your rampack and CMOVE 16K bytes. What's this? It takes more than 16 seconds!

The FORTH words within the DO LOOP are executed 16K times each. Each word makes a CALL to ROM, then returns to FORTH 16K times. There is quite a lot happening. The more FORTH words there are inside the loop the slower it gets, and the loop itself takes time to cycle.

If you define a machine code version of CMOVE, which costs around 26 bytes plus 24 for DEFINER CODE, it will run more than 100 times faster and can shift 100K bytes in less than a second. Incidentally, FORTH-79 has another word CMOVE which decrements addresses, ie moves backwards in memory.

Some programs require large blocks of memory to be searched, sorted, compared or shifted very quickly. In-out ports sometimes need to be scanned at the highest possible speed, and other programs demand the compaction of machine code when there is limited RAM available.

The method of loading machine code outlined in the manual (p147) is particularly suited to transportable machine code FORTH words, which can be used anywhere in your dictionary along with other FORTH words. Another method must be found for loading large blocks of machine code into a fixed area of memory because the C, method of code insertion shown in the manual takes 6 - 8 keystrokes per entry byte.

WHERE TO PUT IT

Remembering the ZX81 (who could forget it?), it has three main areas for storing machine code; in a REM statement at the beginning of program, in an array after screen, and above RAMTOP. The REM and array are saved on tape, but above RAMTOP is not.

The ACE equivalent of a ZX81 REM statement is

```
: MC
  ( ØØØØØØØØØØØØØØØØØØØØØØØØ ad nauseam)
;
```

achieved by holding down the Ø key. If you POKE (Store in FORTH) machine code into the bytes reserved by Ø s, 15454 CALL will run it, assuming you have cleared the dictionary before entering MC. Fortunately the ACE offers an easier method.

```
FORGET the first word in the dictionary and ENTER
CREATE MC
1ØØ ALLOT
```

This will reserve 100 bytes for your machine code starting at 15450. Your dictionary begins at 15441, plus two bytes for the name MC, and another seven for the header, equals 15450, which is easy to remember. You could just as easily have entered 1ØØØØ ALLOT if you have a Rampack, and 10,000 bytes would have been reserved immediately for your ambitious project!

The name of a new FORTH word defined after MC will start at HERE . , which should be 15451 plus the number of bytes reserved by ALLOT. Whenever MC is executed it leaves 15450 on the stack so MC CALL can be used in place of 15450 CALL.

What has actually happened with CREATE MC and ALLOT is that an array has been set up at the beginning of the dictionary. The only reason for putting this array at the beginning of your program is that it gives you free use of REDEFINE and FORGET (as long as you don't REDEFINE or FORGET MC). Whoever devised the word REDEFINE needs a word in the ear! I must have typed this a thousand times since getting my ACE, that would be 9000 keystrokes plus, say, an average of 5000 keystrokes for word names. REDE would have done just as well, saving me 4000 keystrokes thus far. Anyway, if you are careful with REDEFINE and FORGET, MC can be placed anywhere in your dictionary, and MC will put the address of its parameter field, ie the start of your machine code, on the FORTH stack. One of the really good things about FORTH is that it makes it so easy to find addresses, compared with the elaborate PEEKS of ZX BASIC.

Storing machine code above RAMTOP is useful for routines which are needed to stay in RAM, while loading and saving programs, such as programmers' toolkits, data manipulators, etc. The method of reserving an area above Ramtop is implied in the manual (page 141).

```
new address for Ramtop 15384 ! QUIT
```

This can be checked by 15384 @ U. entered from the keyboard.

HOW TO USE IT

The main dissimilarities between machine code on the ZX81 and the ACE are the use of the JP (IY) two byte instruction to return to FORTH (instead of the single byte RET instruction to return to BASIC), and the use of the DE register pair to transfer data from machine code to FORTH (instead of BC to BASIC). RET is used normally with CALL to access machine code subroutines.

With no inbuilt string commands, the ACE needs fewer restart instructions (see pages 147 & 148). Note that RST 24 can be used as a machine code version of DROP if you have temporarily abandoned DE and HL.

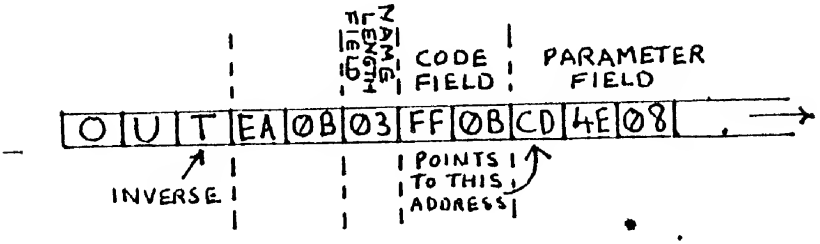
The following are similar in operation:

ACE FORTH	ZX BASIC
address C@ (one byte fetch)	PEEK address
N address C! (one byte store)	POKE address, N
address @ (two byte fetch)	PEEK address + 256 * PEEK address + 1
N address ! (two byte store)	POKE address, N - 256 * INT (N/256) POKE address + 1, INT (N/256)

Some useful instructions are LD HL, (system variable) which transfers the contents of the system variable to your machine code, LD system variable, (HL) which places the contents of HL in the system variable, and RST 8 to print to the screen from the A register. (see figure 1)

LOOKING AT ROM

It is reasonably easy to disassemble FORTH primitives in ROM, using the READ word above. For example, FIND OUT READ entered from the keyboard will show the OUT routine starting at the code field address. OUT has a five byte header:



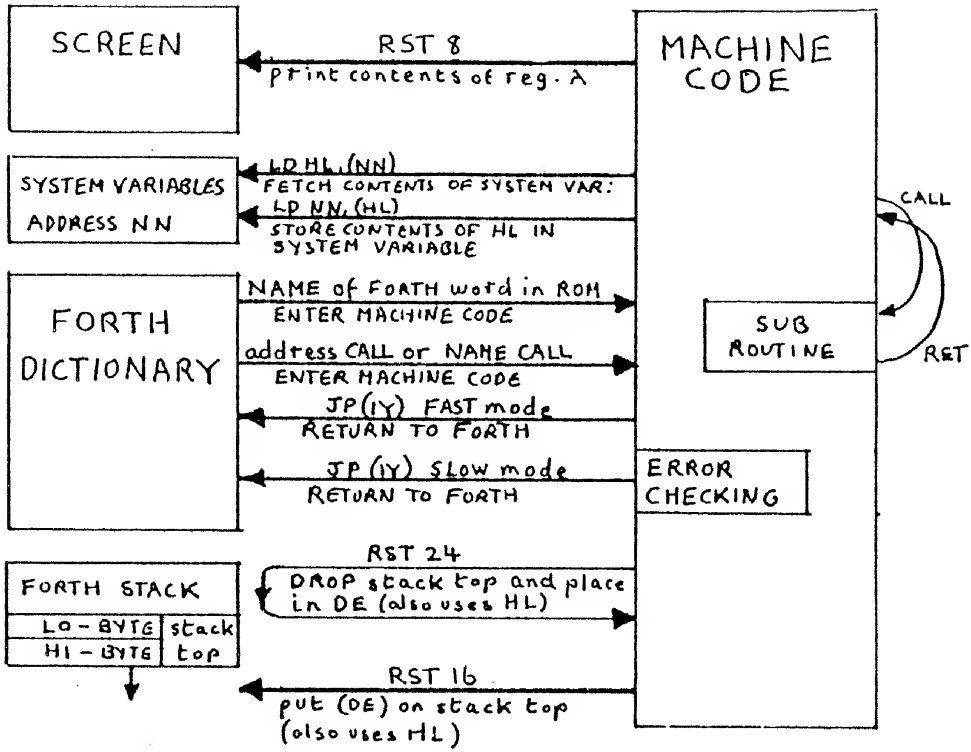


DIAGRAM OF MACHINE CODE ON THE ACG.

HOW TO LOAD IT

There isn't enough memory on the unexpanded ACE to contain a mnemonic assembler, but the alterable number base system makes Hex loading fairly straightforward.

```
: LOADER
  (start address, number of bytes - )
  SWAP DUP ROT + SWAP          (set-up DO LOOP)
  DO
  16 BASE C!                   (into hex)
  QUERY LINE                   (input values and print
  I C! 1 . DECIMAL I . CR      (print hex and decimal
  LOOP                                 addresses)
;
```

This takes the decimal start address and number of bytes you wish to store, accepts a hex input, and prints the result as hex byte, hex address and decimal address. If you want to start with a hex address, enter 16 BASE C! address, number of bytes LOADER. If you prefer to work only in decimal, edit out 16 BASE C! and DECIMAL 1 . To exit from LOADER prior to finishing, do QUIT DECIMAL.

Having stored your machine code, saved it on tape, and found that it only produces a pretty pattern on the screen with a dead keyboard, you will need to find that bug!

HOW TO EDIT IT

The MC bracketed REM shown earlier can be listed as a string of ASCII characters with spaces, which at least shows it is there, but you will probably run into trouble if you try editing in the same way as FORTH words.
You need

```
: READ
  (start address - )
  DUP 21 + SWAP
  DO
  16 BASE C!
  I C@ . I . DECIMAL 1 . CR
  LOOP
;
```

To show 21 consecutive bytes in hex, with hex and decimal addresses. To start, enter decimal address READ. To continue enter new address READ. As with LOADER, you can also start with a hex address, or work only in decimal.

Using LOADER and READ together you can inspect machine code and go back and overwrite it, with only 2-3 keystrokes per entry byte.

Moving backwards and forwards from the code field will be informative. In this case, the parameter field following the code field of OUT starts with CD 4E Ø8. CD is the instruction CALL, which should be followed by a two byte address with the Lo-byte first, thus CD 4E Ø8 is CALL Ø84E. To continue the dissassembly type in 16 BASE C! Ø84E READ which will give

```
Ø84E   2A   LD HL, (3C3B)     (top of Forth stack + 1)
       3B                     (Lo-byte of address)
       3C                     (Hi-byte of address)
       2B   DEC HL            (top of stack address)
       46   LD B, (HL)        (copy Hi-byte of stack
                                top into B)
       2B   DEC HL
       4E   LD C, (HL)        (copy Lo-byte of stack top
                                into C)
       22   LD 3C3B, (HL)     (DROP top number on stack by
                                by resetting stack pointer)
       3B                     (Lo-byte of address)
       3C                     (Hi-byte of address)
       C9   RET               (return to OUT parameter
                                field)
```

FIND OUT 5 + READ (code field address 2 + for Parameter Field and 3 + for CD 4E 08)

```
ØCØ2   DF   RST 24            (put stack top in DE)
       ED                     (prefix)
       59   OUT (C),E         (output contents of E reg
                                (Lo-byte of stack top) to
                                port address in C)
       FD   JP                (jump to the address in IY,
       E9   (IY)               or return to FORTH)
```

3C3B is the system variable SPARE (page 143). Now see where the address ØBEA in the header takes you.

You can disassemble the hex bytes by looking them up in your manual (Appendix A), but if you are not sure about these things, be careful to interpret two and three byte instructions correctly, otherwise you will get lost in a mass of irrelevant and unrelated instructions. Keep a look out for CB and ED prefixes. N signifies that the instruction is followed by a single byte number, NN by a double byte number, and DIS by a single byte displacement. A JP or CALL instruction will tell you to go to another address in ROM. A fuller explanation is well beyond the scope of this article. A good book on Z80 assembly language will help here, as will TAPE 1 PEEKER or TAPE 6 TOOLKIT which show you the word names as well as decimal, hex and binary bytes. If you find machine code difficult to grasp, don't worry, you can do a lot more with FORTH, much faster than with BASIC. How fast and fancy do you need to be?

No doubt someone will come up with a full facility mnemonic dissassembler in a few months from now, when ACE RAM expansion is commonplace.

BIG CHARACTERS RALPH HILTON

Thank you for issue 1 of ACE USER. Here's some additional information on the points covered that you might find of interest.

Entering 16317 HERE - . gives available dictionary space.

On trying out the big letters I found that most of the characters are held in 7 bytes of ROM except those from ASCII 63 to 94 incl. I've attached the listing for printing out the letters which you are welcome to print. Enter BL and then any key pressed will appear on the screen in 8 * normal size (3 lines by 4 columns).

```
: WAIT
  Ø 154Ø3 !
  BEGIN
    154Ø3 @ 2Ø -
  UNTIL
;
Ø variable POS
GR per manual for ACE
: BL
  CLS Ø 7 Ø
  DO
   DUP
  LOOP
  1 GR
  255 7 Ø
  DO
   DUP
  LOOP
  2 GR
  WAIT
  BEGIN
   BEGIN
    INKEY ?DUP
   UNTIL
   15388 @ 8 + POS !
   DUP 32 - 7 * 7554 +
   SWAP DUP
   62 >
  IF
   DUP 94 >
   IF
     DROP 32 - DUP 7 -
   ELSE
     62 - - DUP 6 -
   THEN
  ELSE
    DROP DUP 7 -
  THEN
  DO
    i C@ 15388 @
    7 + 15388 '
    8 Ø
```

```
      DO
        2 /MOD SWAP 1+ EMIT
        15388 @ 2- 15388 !
      LOOP
      DROP 15388 @ 33 + 15388 !
    LOOP
    POS @ DUP 32 MOD Ø=
    IF
      224 +
    THEN
    DUP 9900 >
    IF
      DROP 9216
    THEN
    15388! Ø
  UNTIL
;
```

MISC ROUTINES GARY KNIGHT

1) A simple way of finding out approximately how much memory you have left is to type in 15384 @ 15419 @ - .

2) You can emulate the Basic GET function by using

```
BBGIN INKEY Ø= UNTIL

BEGIN
INKEY
UNTIL
INKEY
```

3) To emulate the Basic Screen$ (X,Y) function, where X is the row and y the column number:—

```
: screen$ 32 * + 9216 + C@ ;
```

Use it by typing in, eg 1Ø 15 SCREEN$
for x=1Ø and y=15

MEMSHOW D.A.WALKER

Many thanks for sending the newsletter and tapes 1 and 2. By way of contribution I enclose my own program MEMSHOW which although now redundant may be of some interest because it does not use PAD, so can be used to illustrate the relevant chapter in the handbook (which I find rather heavy going at times)
Every little helps!

D.A.Walker, London

MEMSHOW (dict: memshow) displays contents of 23 memory addresses from a specified START.
First column addresses in HEX
Second column contents in HEX
Third column characters contents

Unless your first START is in decimal, enter HEX, then starting address then the word START.

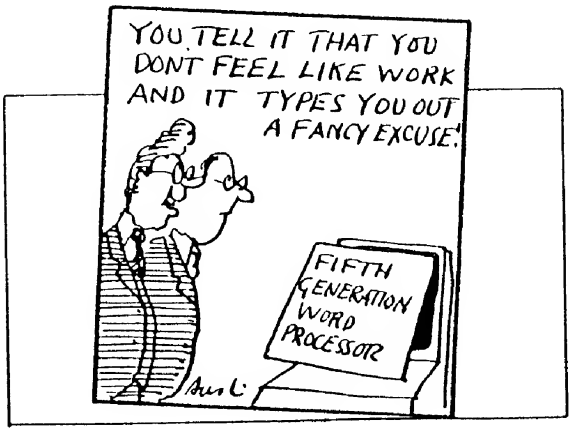
Subsequent starts will be expected in HEX, unless DECIMAL is first entered.

```
: HEX 16 BASE C! ;
(Following definitions in HEX)
: BACK 3CIC(sys. var. SCRPOS) @ 1 - 3CIC ! ;
: 2FORMAT DUP F > IF . ELSE Ø .
BACK . THEN ;

: 4FORMAT DUP FFF >
  IF .
  ELSE Ø . BACK DUP FF >
     IF .
     ELSE Ø . BACK 2FORMAT
     THEN
  THEN ;

: BYTESHOW DUP 4FORMAT 2 SPACES C@
 DUP 2FORMAT 2 SPACES EMIT CR ;

: START HEX INVIS CR 17 Ø DO DUP
 I + BYTESHOW LOOP DROP ;
```



HOLDING FORTH: READERS' LETTERS

KEYBOARD BEEP

While I am very pleased with the ACE, the keyboard can be very annoying, having to press the key carefully and look at the screen each time to see if the key press has registered. My suggestion is for a word to produce a bleep every time a key is pressed. I feel sure that this is possible but all my efforts so far have failed.

R.J.Dunham, Dorchester

If you are detecting single key presses within a program loop you can use BEEP inside IF THEN. For normal entry keyboard bleeps, however, you will need add-on hardware. Where are those circuit designers? Doug

BI-MONTHLY STRINGS

How <u>does</u> one input a string into memory on the Ace?! My only complaint about the ACE (apart from the keyboard - which I replaced with a real one the day I received it) is the manual which, although well-written and produced would have been just that bit better for the inclusion of a few programming examples.
I think the Ace Users Club is going to be a big one. If it does, don't you think a bi-monthly newsletter would be a better format? I'm sure there will be sufficient input "forth"-coming to justify it!

K.Longley, Heywood, Lancs

Yes, a bi-monthly would be better. If there is sufficient demand we shall "Sally Forth" and get busy "Forth-with" (Ouch! -John). To input a string to memory see page 124 and 125 of the Manual, ie use DEFINER STRING and FREDSADDRESS to put strings into the dictionary. Instead of FREDSADDRESS you could use a$, b$, etc. To print the string use a$ type. We are finalising tape 10 which contains a comprehensive collection of words for string handling. Doug. (See also BiblioForth for two articles on strings in Forth. John)

CAN'T READ

Can you tell me why you can't read the CHR$ set memory? The computer obviously can but how? I've got a high(ish) res. program (88 x 64 pixels) which works apart from the fact that you can't read this memory.

Martyn Sudworth, Bristol

The character set in RAM has its bits transposed in such a way as to make it useless for programs. Although it is almost certainly possible to correct this deficiency, it is hardly worth the bother when you have a perfect character set sitting waiting in ROM. Ralph Hilton has come up with a BIG LETTERS listing in this issue which should answer your problem.

(CONT)
As we mentioned in ACE USER no.1, the addresses where characters can be found in ROM are

SPACE (ASCII 32) starts at 1D7A
c (ASCII 127) ends at 1FFB

As Ralph Hilton mentions, characters are held in 7 bytes of ROM except those from 63-94 which are 8 bytes. Doug

FASTER FORTH

I have found that the ACE can print F.P. numbers faster than integer. Try the program below.

TYPE IN: 'SETCLOCK', 'MINSEC' and
'TIME' from the manual (p142)

```
: FASTPRINT CLS
  Ø Ø SETCLOCK
  1ØØØ Ø
  DO
  Ø Ø AT I UFLOAT
  F.
  LOOP
  CR TIME
;
```

I made it 2:06 seconds which is 5 seconds faster than is quoted in the Jupiter Cantab advert.

Ian Briscoe, Maidstone, Kent

We haven't had time to test this. So, no comment. John

MEMORY EXPANSION

I have attempted to expand my Jupiter Ace using a "Memopak" 16K designed for the ZX81. I made an adaptor using an edge connector, a piece of circuit board with 1/10" pitch tracks on both sides, and wiring between the two using the information on p152 and p155 of the ACE manual. At first I was unsuccessful. I tried various combinations of switch positions on the Memopak (presumably these designate the range of addresses to which the Memopak responds) but without success. Following advice from a colleague who has a ZX81 I tried disconnecting A15 (incidently my colleague also checked out the Memopak on his ZX81). At first this appeared to work. For example, 15384 @ U. ENTER gave 32768, showing that the Ace had sensed the extra memory and upped the RAMTOP to 32K. However, there still appears to be a problem, since random errors are occuring. For example, when doing a VLIST, occasional characters are repeated, eg FOORTHH, etc. Also, I have written a test program which writes to and reads from each memory location in turn (avoiding the program area, return stack, etc) looking for differences. There are typically half a dozen such errors in going through the 16K, although their occurence is random. The error is usually of the power of 2, suggesting that one bit is being lost, but it is not always the same bit.

Also, it appears to be a read error rather than a write, as subsequent examination of an error shows the correct value to be stored.
I should certainly be grateful for any advice. I would be interested in your adaptor kit if I thought it would solve the problem, but at the moment I'm not sure that your adaptor is in principle any different from mine.
Roger Edwards, Crawley, W.Sussex

I have a query with regard to the I/O port. I have tried to add the 16K RAM pack of a ZX81 which should work if the wires are changed (as Jupiter say in the manual) but I can't seem to get it to work. Have you any ideas as to why it crashes?
Martyn Sudworth, Bristol

When we tried out our prototype adaptor we had similar problems. We obtained a correct value for Ramtop but not much else! Troubles lay with bad contacts and undervoltages on one of our Rampacks (see article on ACE expansion in this issue). The second prototype worked first time and has been working ever since. Both adaptors are virtually identical! After cleaning the contacts on the first one, it too has been working very reliably.
If your memory expansion stubbornly refuses to work, borrow another Rampack to see if that makes any difference. Also try using a 12v 1.2 amp supply to your ACE in place of the 9v pack (but be carefull not to cause overheating and only do it as a quick test. Do not use a car battery!). If you have a ZX printer or Spectrum power supply, try that. Switch off other electrical appliances in the house to see if they are causing mains interference. What more can we say? We didn't design the Jupiter Ace or ZX81 memory packs, and did not select the type of edge connector used. These things are sent to try us! Doug

Those who just want memory add-on, without any other add-ons, might like to know that Micro Marketing, who wholesale the ACE, are about to market a 16K add-on for the Ace. This will cost about £32 and can be upgraded to 32K for another £10. We hope to have these available through the Users Club. Let me know if you want one. But do bear in mind that they are not of use as they stand, if you want to use sound boxes, ZX Printer, and other ZX add-ons. Also ZX81 memory add-ons are cheaper(due to higher volume production), not to mention all those Ace owners who already have ZX81 memories. John.

BIGGER AND BETTER

I am interested in the ZX81 adaptor, although I did produce a very 'crude' link myself. It proved that the 64K ram pack which I purchased for my ZX81 gives the ACE 48K more memory. I have plans to use the remaining 16K of memory as the display memory for the Thompson CSF Graphic Display Controller EF9366. The chip seems to be very good value for money, even though it does cost the amazing sum of £45 (plus VAT). Thus the type of expansion board which I would like will contain some fairly basic circuitry around this chip and then feed the 64K memory. Programming should be very simple as the chip is controlled through 11 registers

(Z80 ports) and so require only IN/OUT instructions. So our little ACE becomes one A C E of a computer.

Raymond Wand, London SW19

Let us know how you get on. Where will you map the extra 16K? Doug.

PRINTERS

I am presently considering whether to purchase an AMBER 2400 printer to interface to my Jupiter Ace. Do you think this would be a feasible proposition? Have you solved the 'hard copy' problem for LISTing WORD definitions, etc.

David F.Haslam, Stockport

You could use the Amber 2400 with an expanded Jupiter ACE, but you would need suitable software to run it. We think it would be wiser to wait for the parallel interface and other developments in the next few months before choosing a printer. Suggest you contact Amber and see what they have to say. Meanwhile we are still looking at the ZX printer, as well as the Epson range. Doug

BIG DICTIONARY

If you have any information about extending the Ace dictionary in PROM, or of books about Forth programming I would be pleased to know about it.

Robert H.Dunlop, Glenrothes, Fife

See the article on ACE expansion, and BiblioForth column.

BIG CHARACTERS

Thank you for the quick service of the tapes I bought. They load perfectly every time.
Could you please ask Doug how you get big screen characters.

A.Furniss, Ossett, W.Yorks

Thanks for the compliment! See Ralph Hilton's BIG CHARACTERS listing in this issue.

HAM AND CHIPS @ THE ACE FAIR?

I am interested in amateur radio and would welcome any programs for this, ie RTTY, LOG BOOK, CALL SEARCH and QTH LOCATOR. The speed of the ACE would be ideal for radio. I am also interested in chess and other games. I would like to wish you all the best in the new club and that it will run for a long time and have all the support it needs. Who knows you may even have an ACE fair in London!

D.J.Rowe, Eastleigh, Hants

Let's hope so. If it happens the Ace User Group will certainly have a stand there. Any other ACE radio enthusiasts out there?

MARGIN FOR ERROR

Thanks for your helpful notes on tape load/save. There are a couple of points I need to check it seems. If these do not improve the situation I propose to ask the manufacturer for advice/test/comment, on my particular ACE.
Having received your letter and copy of tape this morning, I hope to "play" this evening. Will be interesting to see if "your" tape does better than the Jupiter sample tape. The latter does tend to yield more Error 10s on one side than the other. But that could be due to the small "margin" (Ha-ha) on my set up.

Derek Stenson, Stanmore, Middx

LOAD and SAVE on the Jupiter Ace is certainly better than the ZX81 and is as good as the Spectrum. Until cassette recorder manufacturers make special machines for computers at a reasonable price the problems are likely to remain.
It is better to have a mono cassette recorder without automatic level control on record, and with a sensitive input. As things stand, individual examples of one make of tape recorder vary in terms of maximum input sensitivity and tape-speed. The ACE - like the ZX81 and Spectrum - have very low level outputs which would benefit from low-noise amplification of about 10x.
All the published advice about ZX81 loading problems is applicable to the ACE, including the one of removing the earplug while SAVEing.
Isnt technology wonderful? Doug

ODD BYTES

I see in your first newsletter a word to test for spare dictionary space. Another way to do it which does not produce an error crash is to do the following:

```
: FREE
  15384 @ HERE 12 + - .
;
```

This reads the RAMTOP value and subtracts from it 12 more than the present end of the dictionary. Thus it includes the stack and return stack as available space included with the free space. You can allow for the odd 10-20 bytes these stacks need when seeing if your programme will fit. This word can be loaded before or after any application programme so you can easily see how much space it takes up. On an empty dictionary (apart from FREE) the answer is 900 bytes exactly.

Alan Buckman, Deal, Kent

This method gives a slightly optimistic byte count and does not take account of the error check margin before the stacks collide. The difference is approximately 24 bytes. Doug

DRIVES FORTH

Is it possible for the JUPITER ACE to fit the ZX SPECTRUM disc drives? If so could you send information.

M.P.Perkins, Reading, Berks

It should be possible, in the fullness of time, to use micro floppies and discs on the ACE, when the hardware people get around to looking at the problem. It has been predicted that a ZX81 conversion will be available for the SPECTRUM microdrive within months of it appearing. I suppose it all depends on the number of Jupiter Aces sold, which is again dependent on the range of software and peripherals available. Chicken and egg! With home computers you have to learn patience, as well as a language or three. Doug (Pascal next, Doug? John)

FORTH WAVE?

I have recently brought a Jupiter Ace and would like to join the Users' Club.
Although I will regretably not be able to take an active part in the Club, due to my profession as a Naval Officer keeping me at sea for the next year or two, I look forward to reaping the benefits of add-ons/software offered by the club.
Details of more memory and I/P O/P ports would be most appreciated.

R.K.Ellis, Lieutenant, Royal Navy

See article on ACE expansion.

BOOKS

BRODIE, Ian. Starting Forth. Prentice-Hall.

The US classic. Designed with disc-based systems in mind. Also nothing on strings. Definately overpriced in Britain.

DEGRANDIS-HARRISON, Richard. Forth theory and practice. Acornsoft, 1982. 2nd ed.

Forth for the Acorn Atom. Not FORTH-79 but fairly comprehensive.

KNECHT, Ken. Introduction to Forth. Sams(USA), 1982.

Based on MMSFORTH for TRS-80 Models I and III. Doesn't deal with FORTH-79. Good on strings.

VICKERS, Steven. Forth programming. Jupiter Cantab, 1982. 2nd ed.

Forth for the Jupiter Ace. A Spanish edition is in preparation from Sushiro Data, Barcelona, which will include extra programming examples.

PERIODICALS

The notes that follow are based on a visit to the Science Reference Library, Chancery Lane, London, together with a scanning of the British microcomputer magazines we keep here at Remsoft.

FORTH DIMENSIONS is the magazine of the US Forth Interest Group (POB 1105, San Carlos, CA 94070, USA). Unfortunately I've yet to locate a set in Britain. Can anyone help?

The FORTH INTEREST GROUP (UK) also has a newsletter, which I haven't seen either. To contact FIG-UK write to the Honorary Secretary, Forth Interest Group, 15 St Albans Mansion, Kensington Court Palace, London W8 5QH. An S.A.E. would be courteous.

BYTE (McGraw-Hill, USA) is the premier US micro magazine, and is also available in Britain through WH Smiths and the like. They featured FORTH in the August 1980 issue (vol.5 no.8) with several other articles in following issues. What follows is a list of FORTH articles in BYTE:

Charles H.MOORE. The evolution of FORTH, an unusual language. BYTE Aug 80: 76-92
The inventor of FORTH outlines its history.

John S. JAMES. What is FORTH?: a tutorial introduction. BYTE Aug 80: 100-126
Good introduction, with an excellent bibliography.

A.Richard MILLER & Jill MILLER. Breakforth into FORTH! BYTE Aug 80: 150-162
  The authors of MMSFORTH for the TRS-80 on FORTH for games on the TRS-80 Model I.

Kim HARRIS. FORTH extensibility: or how to write a compiler in 25 words or less. BYTE Aug 80: 164-184
  Based on FORTH-79.

Gregg WILLIAMS. FORTH glossary. BYTE Aug 80: 186-196
  Glossary of FORTH words used in articles in this issue of BYTE.

William RAGSDALE. The FORTH Standards Team. BYTE Aug 80: 274-277
  The evolution of FORTH-79.

Selected FORTH vendors. BYTE Aug 80: 98
  With addresses.

John CASSADY. Stacking strings in FORTH. BYTE Feb 81: 152-162
  Extra words for string-manipulation to supplement FORTH-79.

Ulrich FREI. KNIGHT: a knight's tour problem in MMSFORTH. BYTE Feb 81: 325
  Chess puzzle for TRS-80 Model I.

John BUMGARNER. A coding sheet for FORTH. BYTE March 81: 155-162
  Looks useful - with blank form, and worked example.

Valo G MOTALYGO. PS - a FORTH-like threaded language, Part 1. BYTE Oct 81: 462-466
  An attempt to improve FORTH assembler, with a new language.

Valo G.MOTALYGO. PS - a FORTH-like threaded language, Part 2. BYTE Nov 81: 400-408
  Conclusion of this article with example and PS Glossary.

Allyn RICHARDSON. The Datahandler from Miller Microcomputer Services. BYTE Nov 81: 138-150
  Review of a personal data-base management and utilities program written in MMSFORTH for the TRS-80. Disc-based for Model 1, with minimum of 32K.

DR.DOBB'S JOURNAL is a US magazine which has consistently championed FORTH, particularly because FORTH itself was put into the public domain by its inventors. The Science Reference Library's set of this journal is incomplete. What follows is a list of articles noted on my recent visit:

Ralph DEANE. A proposal for strings for FORTH. DR.DOBB'S JOURNAL no.50, Nov-Dec 80: 40-45

H.T.GORDON. What FORTH is? DR.DOBB'S JOURNAL no.51, Jan 81: 8-15
  Introduction; correction in no.53:5; letters in nos.53:5, 54:4,6; 55:4,6.

H.T.GORDON. PARFOR: a theoretical, parametrized FORTH-like HLL. DR.DOBB'S JOURNAL no.52, Feb 81:8-13
Esoteric (HLL- High Level Language).

S.B.BASSETT. Algorithms in FORTH. DR.DOBB'S JOURNAL no.54, April 81: 16-17, 20

Special FORTH issue, DR.DOBB'S JOURNAL no.59, Sept 81:

Kim HARRIS. The FORTH philosophy. p6-11

William F.RAGSDALE. A FORTH assembler for the 6502. p13-16,18-24

R.G.LOELIGER. Sallying FORTH to battle. p25-26,28

Henry LAXEN. Screen-oriented editor in FORTH. p27-41
With listing. Author wrote the editor for the HHC hand-held computer.

Glenn B.HAYDON. Elements of a FORTH data base design. p42-45,47,56

Ray DUNCAN. FORTH decompiler. p48,50-53

Mitchell E.TIMIN. The FORTH alternative. p57-59
Introductory description of FORTH from the author of TiminFORTH, which runs on 8080 or Z80 systems with CP/M and at least 24K memory.

John S.JAMES. Program for a modem: FORTH code from the conference tree. DR.DOBB'S JOURNAL no.67, May 82:18-24
Includes listings for modem procedures. Could be useful!

THE BEST OF THE COMPUTER FAIRES vol.4: Conference Proceedings of the 4th West Coast Computer Faire, San Francisco, 1979.
Systems Software: Forth. p296-318.
A series of short articles by members of the US Forth Interest Group. Could be worth a look.

Science Reference Library, Chancery Lane (Holborn branch): locations:

BYTE: (P)PP17-E(18)

DR.DOBB'S JOURNAL: (P)PM76-E(25)
COMPUTER FAIRES: (P)PM76-E(8)

Unbound recent issues in basement.Bound volumes on top gallery.

Very good range of US micro magazines, esp in computers and music, computers & art, etc. Also the Br. & US trade mags that you dont see on the newsagents shelves.

Whilst you're there have a look in the SMALL COMPUTER PROGRAM INDEX (basement - (P)PM76-E(47). Nothing under FORTH in the ones I looked under, but I was in rather a hurry ...

There is a new battery-powered microcomputer called the HHC, marketed by Panasonic and Quasar in the USA, and being test-marketed in Britain and Europe by Olympus, which runs SNAP, a FORTH-like language. Looks interesting. There's an article on this micro in BYTE Jan 81: 34-45.

In Britain there have been few articles on FORTH in the popular microcomputing magazines. Exceptions are:

N.K.FREESTONE. FORTH computes OK. ELECTRONICS & COMPUTING MONTHLY vol.3 no.9, Sept 82: 22-24
Useful introduction.

N.K.FREESTONE. FORTH computes OK. Part 2. ELECTRONICS & COMPUTING MONTHLY vol.3 no.10, Oct 82: 24-27
This two-part article is based on the author's knowledge of FORTH on the Hewlett-Packard HP2100.

D.S.PECKETT. Going FORTH. COMPUTING TODAY Jan 82:45-48
Pt 1 of a series. Based on MMSFORTH which runs on 16K TRS-80 or Video Genie.

D.S.PECKETT. Going FORTH. COMPUTING TODAY Feb 82:61-64
Pt 2. Defining new words, and comparative testing.

D.S.PECKETT. Going FORTH. COMPUTING TODAY March 82: 48-51
Pt 3. Programming.

D.S.PECKETT. Going FORTH. COMPUTING TODAY April 82: 91-95
Pt 4. Explanation and listing of a Towers of Hanoi program. Anyone tried it out for the Ace? We'd be interested to hear from you.

J.YALE. Factorials. PRACTICAL COMPUTING Sept 82: 151
FORTH program for Factorials. Anyone tried it on the Ace?

David SANDS. As you learn FORTH, it learns from you. PRACTICAL COMPUTING Aug 81: 92-94
A review of Stackworths FORTH, running under CP/M on the North Star micro.Plus a description of the author's own FORTH for the North Star, implemented in Z-80 language.

Articles in British magazines on ACEFORTH are mainly of the let's-review-a pre-production-copy-type. However a few listings have appeared:

Simon CROSS. Sailors' Hornpipe. POPULAR COMPUTING WEEKLY December 16, 1982:22
The author of our Tape 9, which has good sound, so this program may be worth a look. Anyone tried it?

Martyn SUDWORTH. Jupiter Ace revisited. POPULAR COMPUTING WEEKLY January 20, 1983:22-23
Another Club member. Gives a listing for a 3K game called Alien Swarm. Anyone tried it?

Martin SUDWORTH. Music on Jupiter Ace. POPULAR COMPUTING WEEKLY January 27, 1983: 15
Short program for making music. (The different spellings of Martin/Martyn's name is courtesy of PoCW, not me!)

Andrew CURTIS. Chase on Jupiter Ace. POPULAR COMPUTING WEEKLY January 27, 1983: 19
Game listing. Curiously uses one word for entire program, (or is it the typesetter?)

IF I'VE MISSED ANY ARTICLES OR LISTINGS, DO WRITE IN!

TAPE 1: PEEKER (3K) — DOUG BOLLEN

Unravel Rom and Ram, 20 bytes per screen in decimal, hex, character, binary byte. Addresses in decimal or hex.
CLUB PRICE: £3.50; OTHERS: £4.50

TAPE 2: NIGHT RIDER/SKETCH/EDITOR (3K) — DOUG BOLLEN

NIGHT RIDER: You are driving along a twisty road at night. Can you stay in the left hand lane and score points?
SKETCH: Draw horizontal, vertical, or diagonal lines with any keyboard character or graphics. Rubout and cursor adjust. Save your picture on tape, independent of the program. Quickly design your own screen layout. Pictures can be altered.
EDITOR: Transfer text from input buffer to screen with cursor position control. Erase letters or lines and replace. File screens of text on tape independent of the program. Search for and load screens by name, and update.
DEMO: Simple screen demo to demonstrate bload and bsave.
CLUB PRICE: £4.50; OTHERS:£5.50

TAPE 3: 3 FAST GAMES (3K) — IAN BRISCOE

APPLEATER: Eat as many apples as you can without falling down the hole, or being caught by the monster. The monster eats your apples, but you can trap it into falling down the hole to get another game. Fast action!
PARACHUTE: Prevent the enemy from landing by shooting the parachutists, but if they land, the game ends. Part machine code.
METEOR CRUISE: Avoid the meteors and spear the loops to score. Can you better 280? Part machine code.
CLUB PRICE: £4.50; OTHERS: £5.50

TAPE 4: 4 CHALLENGING GAMES (3K) — GARRY KNIGHT

SAUCER: You must wipe out the saucers, losing a life for each one that passes you. You have nine lives.
DRIVER: There are penalty points for not keeping your car on the road.
MAZE: Sets up a semi-random maze, the idea being to reach the money at the top right of the screen as quickly as possible. Press any key to replay.
PARA: A plane flies across the screen again and again, until you press the 7 key, then the parachutist drops. You must guide him onto the pad, but there is a very gusty wind blowing, so it's not easy. Any key to continue.
CLUB PRICE: £4.50; OTHERS: £5.50

TAPE 5: FROGGER (19K) — GARRY KNIGHT

Hop your frog across a busy four lane road to the safe centre island. From the island you must hop from log to log to one of four safe backwaters. Don't get run over, don't fall in the water, don't slip off the end of a log. The game speeds up as your points increase. Quite fast this one.Uses 5.3K. CLUB PRICE: £4.50; OTHERS: £5.50

TAPE 6: TOOLKIT/SCREENKIT (3K) GARRY KNIGHT

TOOLKIT: Contains a short list of words designed to aid ROM disassembly and program development. Scan memory contents in decimal, hex, character, and binary, with decimal and hex addresses. Scroll forwards or back. Poke a new value into selected address. Spare memory bytes. Definer word for your machine code routine.
SCREENKIT: Machine code screen handling routines. Scroll up or down. Scroll left or right with or without wrap. Invert. Fill. Change occurence of character.
DEMO: Shows the speed of machine code, with above routines.

CLUB PRICE: £6.50; OTHERS: £7.50

TAPE 7: CHARACTER DESIGNER (19K) GARRY KNIGHT.

Design characters quickly and easily with this aid. Store your character in character set ram. Lists the eight numbers used in decimal, hex, and binary.

CLUB PRICE: £5.50; OTHERS: £6.50

TAPE 8: PICASSO (19K) GARRY KNIGHT

Draw and store pictures in memory using pixels.

CLUB PRICE: £4.50; OTHERS: £5.50

TAPE 9: 3 GAMES WITH GOOD SOUND (3K) SIMON CROSS

BOMBER: A spaceship passes from left to right above a city of skyscrapers, it loses height with each crossing. You are in the spaceship and need to land safely, the only way being to drop bombs on the city destroying it to produce a flat landing strip. User-selectable speed.
GORGER: A truly original program. Can you beat 49?
BOUNCY: A version of the perennial Breakout, featuring user-defined graphics, sound, on-screen scoring and user-selectable speed.

CLUB PRICE: £4.50; OTHERS: £5.50

TAPE 10: STRINGS (19K) GARRY KNIGHT

40 Forth words for String Handling (in less than 2¼K). Includes CMOVE, EXPECT, COUNT, AND, -TRAILING.
10K of user dictionary. 12K of instructions. We're very proud of this one!

CLUB PRICE: £6.50; OTHERS: £7.50

ADAPTOR FOR INTERFACING ZX81 ADD-ONS (KIT)

We supply the parts, you supply the solder and labour! Comes with full instructions.
Subject to parts availability(up to 4 weeks).

CLUB MEMBERS ONLY: £7.50

JUPITER ACE MICRO COMPUTER

CLUB MEMBERS ONLY: £84.95

ALL PRICES INCLUDE POSTAGE & PACKING. DELIVERY USUALLY WITHIN 21 DAYS.

THE NATIONAL JUPITER ACE USERS CLUB

ANNUAL SUBSCRIPTION: £7 UK
£8 ELSEWHERE (PAYMENT IN STERLING)

MEMBERS GET THE NEWSLETTER 'ACE USER' AT LEAST THREE TIMES A YEAR, PLUS A DISCOUNT OF £1 PER TAPE ON SOFTWARE, AND SPECIAL OFFERS ON HARDWARE.

SAE FOR DETAILS TO

REMSOFT, 18 GEORGE ST, BRIGHTON BN2 1 RH

OR PHONE JOHN NOYCE ON 0273-602354

(CHEQUES PAYABLE REMSOFT PLEASE)

HERE ENDETH THE SALES BLURB ...

Cartoons by David Austin: reprinted from 'Microchips with everything: the consequences of information technology' edited by Paul Sieghart, and published by the Comedia Publishing Group, 9 Poland Street, London W1V 3DG, price £3.50 paperback.


IT'S ULTRA REALISTIC. IF YOU WIN YOU GET A PRIZE YOU DIDN'T WANT IN THE FIRST PLACE!
FALKLANDS FLEET
ELECTRONIC GAME


THE OUTPUT FROM OUR RACING FORM PROGRAM IS PARTICULARLY USER-FRIENDLY.